

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ "ЛЬВІВСЬКА
ПОЛІТЕХНІКА"**

Кафедра систем штучного інтелекту

Лабораторна робота № 5

з дисципліни

“Дискретна математика”

Виконав:

студент групи КН-114

Гудима Анастасія

Викладач:

Мельникова Н.І.

Львів - 2019

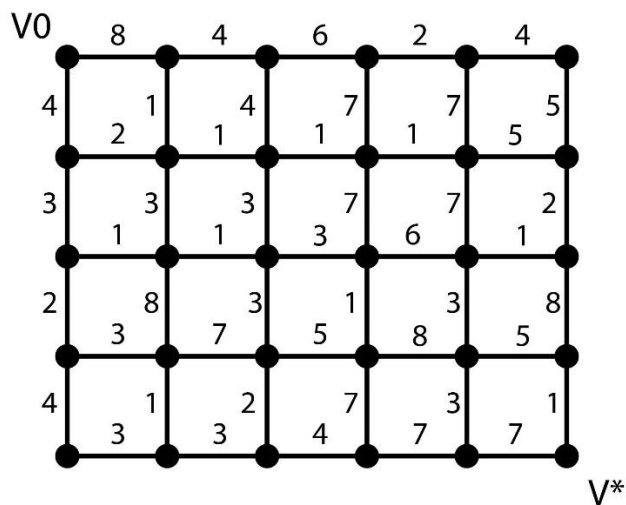
Тема. Знаходження найкоротшого маршруту за алгоритмом Дейкстри. Плоскі планарні графи.

Мета. Набуття практичних вмінь та навичок з використання алгоритму Дейкстри.

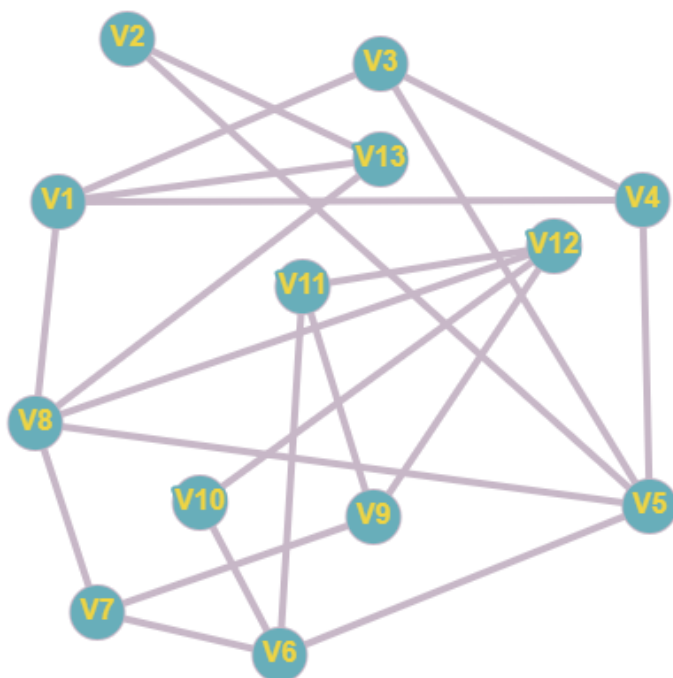
Завдання варіанту №5 з додатку 1

Розв'язати на графах наступні 2 задачі:

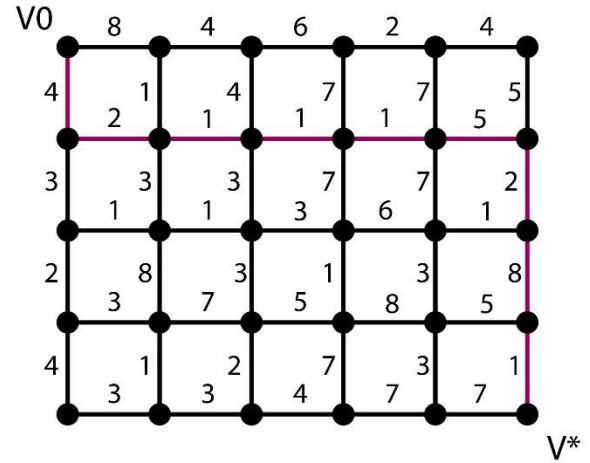
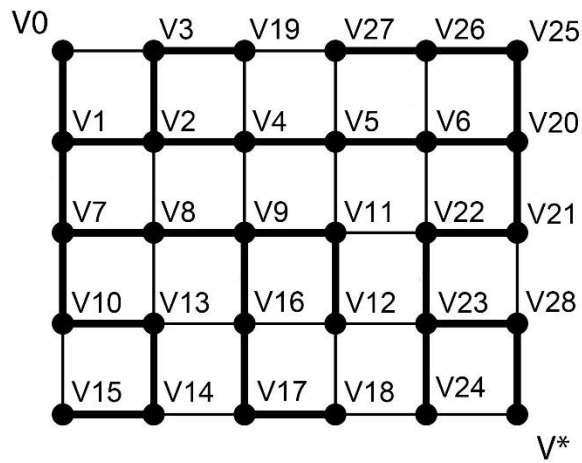
1. За допомогою алгоритму Дейкстри знайти найкоротший шлях у графі поміж парою вершин V_0 і V^* .



2. За допомогою γ -алгоритма зробити укладку графа у площині, або довести що вона неможлива.



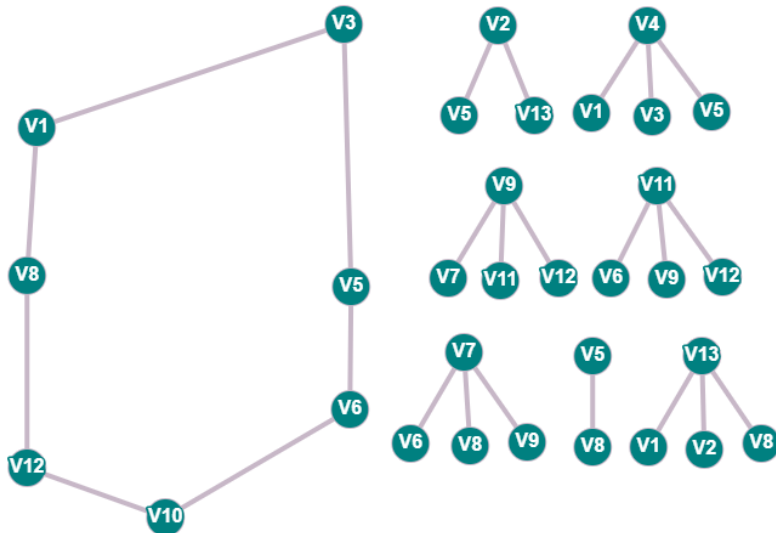
Розв'язок



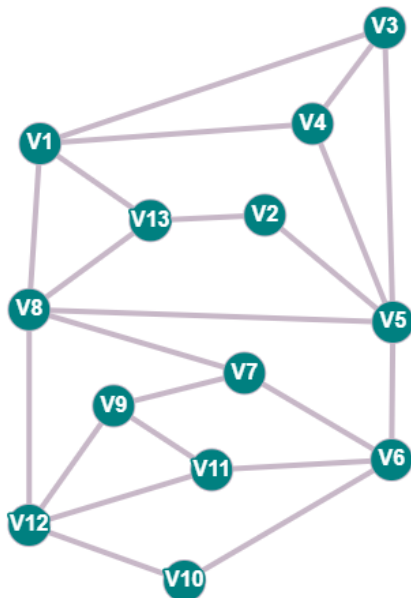
1.

Шуканий найкоротший ланцюг: $[V0, V1, V2, V4, V5, V6, V20, V21, V28, V^*]$.

Довжина ланцюга: $L=L(V^*)=25$.

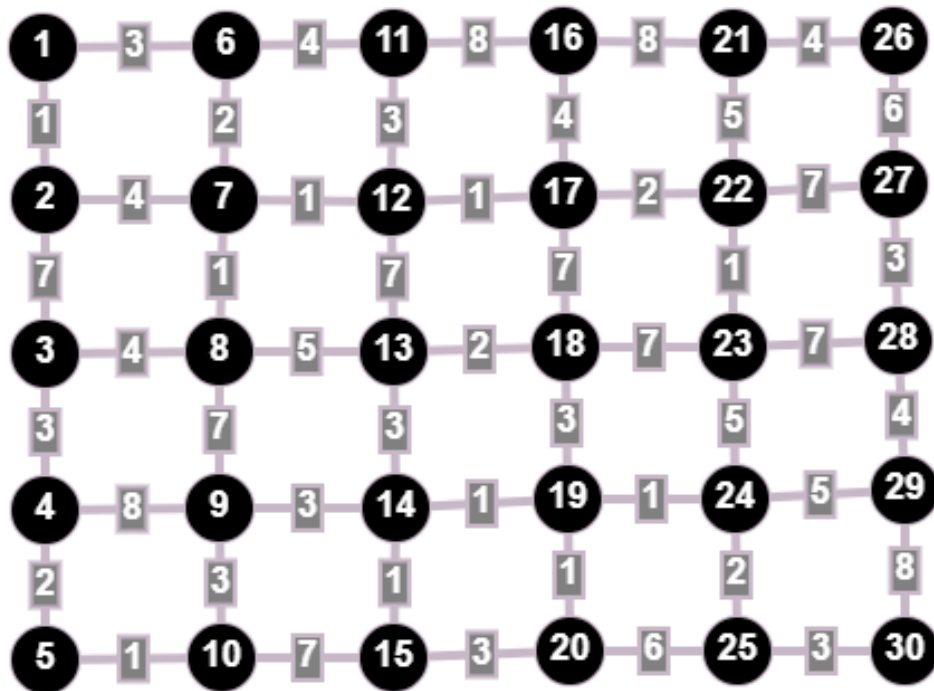


2.

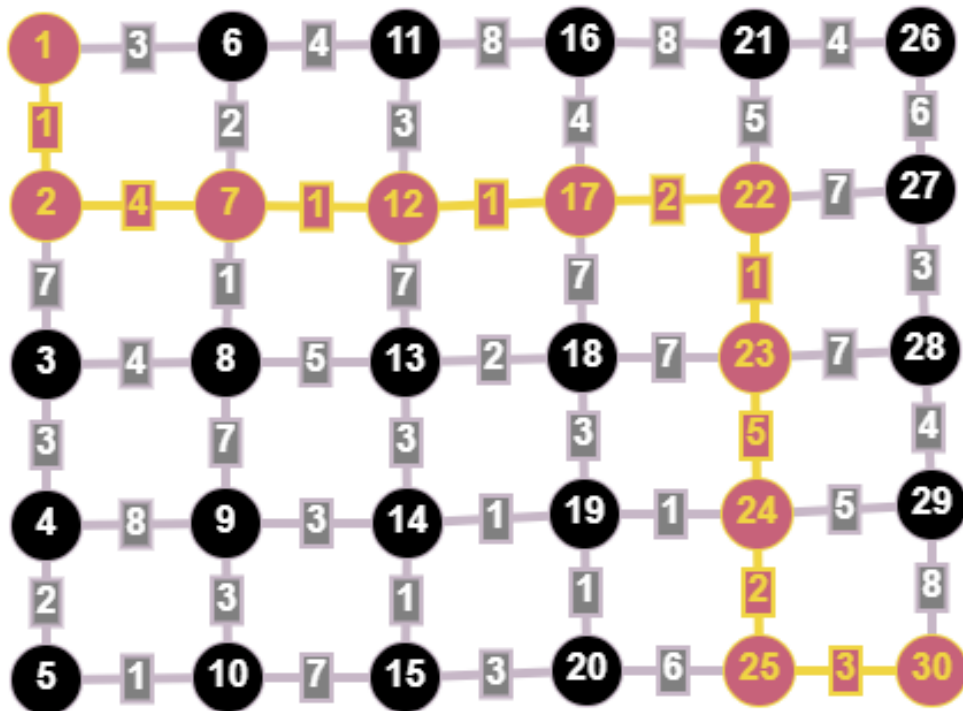


Завдання варіанту №5 з додатку 2

Написати програму, яка реалізує алгоритм Дейкстри знаходження найкоротшого шляху між парою вершин у графі. Протестувати розроблену програму на графі згідно свого варіанту.



Розв'язок



Довжина ланцюга від 1 вершини до 30: $L=20$.

```

#include <iostream>

using namespace std;

int main()
{
    int n,m;
    cout<<"How many vertexes do you want to add? ";
    cin>>n;
    cout<<"How many edges do you want to add? ";
    cin>>m;
    cout<<endl;

    int s=n;
    int a[s][s];
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<n; j++)
        {
            a[i][j]=0;
        }
    }

    //вводимо потрібні ребра
    int t1,t2,w;
    cout<<"Enter edges(first vertex | second vertex | weight):\n";
    for (int k=0;k<m;k++)
    {
        cout<<"Edge"<<k+1<<": ";
        cin>>t1>>t2>>w;
        while (t1<0 || t1>n || t2<0 || t2>n || w<0)
        {
            cout<<"The data entered incorrectly."<<endl;
            cout<<"Try again please."<<endl;
            cin>>t1>>t2>>w;
        }
        a[t1-1][t2-1]=w;
        a[t2-1][t1-1]=w;
    }
    cout<<endl;

    int sh[n]; //мін. шляхи
    int v[n]; //пройдені вершини
    int temp, index, minn;
    for (int i = 0; i<n; i++)
    {
        sh[i] = 300000;
        v[i] = 1;
    }
    sh[0] = 0;

    do
    {
        index = 300000;
        minn = 300000;
        for (int i = 0; i<n; i++)
        {
            //проходимо всі вершини
            //якщо вершина не пройдена і вага менша 300000
            if ((v[i] == 1) && (sh[i]<minn))
            {

```

```

        minn = sh[i];
        index = i;
    }
}

//додаємо знайдену мінімальну вагу і порівнюємо
if (index != 300000)
{
    for (int i = 0; i < n; i++)
    {
        if (a[index][i] > 0)
        {
            temp = minn + a[index][i];
            if (temp < sh[i])
            {
                sh[i] = temp;
            }
        }
    }
    v[index] = 0;
}
} while (index < 300000)

cout<<"Shortest length from V1 to V"<<n<<" is: "<<sh[n-1]<<endl;

//пошук шляху
int vvv[n];
int kin = n-1;
vvv[0] = kin + 1;
int k = 1; //індекс попередньої вершини
int weight = sh[kin];

while (kin != 0) //поки не дійдем до поч. вершини
{
    for (int i = 0; i < n; i++)
    if (a[kin][i] != 0) //перевірка чи є таке ребро
    {
        int temp = weight - a[kin][i]; //шукаємо вагу попередньої вершини
        if (temp == sh[i])
        {
            weight = temp;
            kin = i;
            vvv[k] = i + 1;
            k++;
        }
    }
}

//вивід шляху
cout<<"\nThe shortest way:\n";
for (int i = k - 1; i >= 0; i--)
{
    cout<<vvv[i]<<" ";
}
cout<<endl;

return 0;
}

```

```
How many vertexes do you want to add? 30
How many edges do you want to add? 49

Enter edges(first vertex | second vertex | weight):
Edge1: 1 2 1
Edge2: 1 6 3
Edge3: 2 3 7
Edge4: 2 7 4
Edge5: 3 4 3
Edge6: 3 8 4
Edge7: 4 9 8
Edge8: 4 5 2
Edge9: 5 10 1
Edge10: 6 11 4
Edge11: 6 7 2
Edge12: 7 12 1
Edge13: 7 8 1
Edge14: 8 13 5
Edge15: 8 9 7
Edge16: 9 14 3
Edge17: 9 10 3
Edge18: 10 15 7
Edge19: 11 16 8
Edge20: 11 12 3
Edge21: 12 17 1
Edge22: 12 13 7
Edge23: 13 18 2
Edge24: 13 14 3
Edge25: 14 19 1
Edge26: 14 15 1
Edge27: 15 20 3
Edge28: 16 21 8
Edge29: 16 17 4
Edge30: 17 22 2
Edge31: 17 18 7
Edge32: 18 23 7
Edge33: 18 19 3
Edge34: 19 24 1
Edge35: 19 20 1
Edge36: 20 25 6
Edge37: 21 26 4
Edge38: 21 22 5
Edge39: 22 27 7
Edge40: 22 23 1
Edge41: 23 28 7
Edge42: 23 24 5
Edge43: 24 29 5
Edge44: 24 25 2
Edge45: 25 30 3
Edge46: 26 27 6
Edge47: 27 28 3
Edge48: 28 29 4
Edge49: 29 30 8

Shortest length from V1 to V30 is: 20

The shortest way:
1 2 7 12 17 22 23 24 25 30

Process returned 0 (0x0)    execution time : 31.406 s
Press any key to continue.
```

Висновок: впродовж виконання лабораторної роботи я набула практичних вмінь та навичок з використання алгоритму Дейкстри.